Church, Cardinal and Ordinal Representations of Integers and Kolmogorov complexity

Marie Ferbus-Zanda, Serge Grigorieff

Church, Cardinal and Ordinal Representations of Integers and Kolmogorov complexity

Denis Richard's 60th Biirthday Conference

Marie Ferbus-Zanda
Université Paris 7
2, pl. Jussieu 75251 Paris Cedex 05
France
`ferbus@logique.jussieu.fr`

Serge Grigorieff
LIAFA, Université Paris 7
2, pl. Jussieu 75251 Paris Cedex 05
France
`seg@liafa.jussieu.fr`

Mai 2002

Contents

Abstract

We consider classical representations of integers: Church's function iterators, cardinal equivalence classes of sets, ordinal equivalence classes of totally ordered sets. Since programs do not work on abstract entities and require formal representations of objects, we effectivize these abstract notions in order to allow them to be computed by programs. To any such effectivized representation is then associated a notion of Kolmogorov complexity. We prove that these Kolmogorov complexities form a strict hierarchy which coincides with that obtained by relativization to jump oracles and/or allowance of infinite computations.

1 Kolmogorov complexities

We shall use the following notations.

Notation 1.1.
1) Inequality, strict inequality and equality up to a constant between functions $\mathbf{N} \to \mathbf{N}$ are denoted as follows:

$$\begin{array}{lcll} f \leq_{\mathrm{ct}} g & \Leftrightarrow & \exists c\, \forall n\, (f(n) \leq g(n) + c) & \\ f <_{\mathrm{ct}} g & \Leftrightarrow & f \leq_{\mathrm{ct}} g \ \wedge\ \forall c\, \exists n\, (f(n) < g(n) - c) & \\ f =_{\mathrm{ct}} g & \Leftrightarrow & \exists c\, \forall n\, (|f(n) - g(n)| \leq c) & \Leftrightarrow\quad f \leq_{\mathrm{ct}} g \ \wedge\ g \leq_{\mathrm{ct}} f \end{array}$$

2) Y^X (resp. $X \to Y$) denotes the set of total (resp. partial) functions from X into Y.

3) We denote $\varphi_{\mathtt{e}}$ the partial recursive function $\mathbf{N} \to \mathbf{N}$ with code $\mathtt{e}$.

4) We denote $card(X)$ the number of elements of X in case X is a finite set.

1.1 Kolmogorov complexity

Definition 1.1 (Kolmogorov, 1965 [5]).
Kolmogorov complexity $K : \mathbf{N} \to \mathbf{N}$ is defined as follows:

$K(n)$ is the shortest length of a program which halts and outputs n

To make Def. 1.1 meaningful, some points have to be precised:

(Q1) *Where are programs taken from? In which alphabet?*
Bigger the alphabet, shorter the programs. We shall therefore fix the alphabet of all programming languages to be binary. Now, Kolmogorov's invariance theorem insures that there exist optimal universal programming languages U such that, for any programming language V, the associated complexity functions K_U, K_V satisfy $K_U \leq_{\mathrm{ct}} K_V$ (cf. Notations 1.1). In particular, if U_1, U_2 are two optimal universal programming languages then $K_{U_1} =_{\mathrm{ct}} K_{U_2}$.

(Q2) *What does it mean that a program outputs an integer n ?*
A program can only output a formal object such as a word in some alphabet which represents n. However, there is again an invariance property relative to the usual representations of the output integer n : up to a constant, the same complexity functions are obtained when considering unary representation or base k representation for $k \geq 2$ (cf. [7] or [4] or [10]).

The question aroused by $(Q2)$ is the core of this paper. We shall reconsider it in §2 and §3.

1.2 Infinite computations and oracles

Chaitin, 1976 [3], and Solovay, 1977 [11], considered Kolmogorov complexity of infinite objects (namely recursively enumerable sets) produced by infinite computations.
Allowing programs leading to possibly infinite computations but finite output (i.e. remove the sole halting condition) , we get a variant of Kolmogorov complexity for which the results mentionned in $(Q1)$ above also apply.

Definition 1.2.
Allowing programs with possibly infinite computations, Kolmogorov complexity $K^\infty : \mathbf{N} \to \mathbf{N}$ is defined as follows:

> $K^\infty(n)$ *is the shortest length of a (possibly non halting) program which outputs* n *in unary representation*

Remark 1.1.
The definition of $K^\infty(n)$ is dependent on the unary representation of outputs: $(Q2)$ does not apply.

Kolmogorov complexity can also be considered for computability relative to some oracle.

Definition 1.3.
Considering partial recursiveness relative to some oracle A, Def. 1.1, 1.2 lead to relative Kolmogorov complexities $K^A : \mathbf{N} \to \mathbf{N}$ and $K^{A,\infty} : \mathbf{N} \to \mathbf{N}$. In case $A = \emptyset^i$ is the i-th jump (i.e a Σ^0_i-complete set of integers), we simply write $K^i, K^{i,\infty}$.
For explicit values $i = 1, 2, \ldots$ we also write $K', K'' \ldots, K'^\infty, K''^\infty \ldots$.

Jump oracle $\emptyset'$ (resp. $\emptyset''$,...) allows the computation to decide for free (in a single computation step) any Σ^0_1 or Π^0_1 (resp. Σ^0_2 or Π^0_2,...) statement about integers. As expected and is well known, such an oracle leads to an extended notion of programs. Which allows to

- compute non recursive sets and functions,
- get much shorter programs to compute finite objects or recursive sets and functions, namely $K >_{\mathrm{ct}} K' >_{\mathrm{ct}} K'' >_{\mathrm{ct}} \ldots$.

Infinite computations also lead to shorter programs but, as proved by Becher, 2001 [1], they do not help as much as the jump oracle.

Proposition 1.1 ([1]).

$$K >_{\mathrm{ct}} K^\infty >_{\mathrm{ct}} K' >_{\mathrm{ct}} K'^\infty >_{\mathrm{ct}} K'' >_{\mathrm{ct}} \ldots$$

1.3 Prefix Kolmogorov complexities

Introduced by Levin [6] and Chaitin [2] (cf. [7]), prefix complexity $H : \mathbf{N} \to \mathbf{N}$ is the analog of Kolmogorov complexity obtained by restricting programming languages to be prefix-free: two distinct programs have to be incomparable with respect to the prefix ordering. Variants $H', H'' \ldots, H^{\infty}, H'^{\infty}, \ldots$ involving infinite computations and/or relativization are defined in the obvious way.

As concerns all questions considered in this paper, everything goes through with straightforward changes. So that we shall deal exclusively with the K complexity and its variants $K^{\infty}, K', K'^{\infty}, K'', \ldots$.

2 Representations of integers

The purpose of the paper is to consider particular representations of integers and to study their influence on the definition of Kolmogorov complexity as pointed in question ($Q2$) relative to Def. 1.1 above. This will lead to a strict hierarchy of Kolmogorov complexities which happens to coincide with that obtained in Prop. 1.1.

2.1 Abstract representations of integers

A representation of integers involves some abstract object C (in practice much more complex than $\mathbf{N}$ itself) such that some of its elements characterize the diverse integers through some property. Each representation illuminates some role and/or properties of integers.

Definition 2.1.
A representation of integers is a pair (C, R) where C is some (necessarily infinite) set and R is a *surjective* partial function $R : C \to \mathbf{N}$.

Remark 2.1.
In practice, $Domain(R)$ will be a strict subset of C, in fact a very small part of C.

Example 2.1.
1) The unary representation of integers corresponds to the free algebra built up from one generator and one unary function, namely 0 and the successor function $x \mapsto x + 1$.

2) The various base k (with $k \geq 2$) representations of integers also involve term algebras, not necessarily free. They differ by the sets of digits they use but all are based on the usual interpretation $d_n \ldots d_1 d_0 \mapsto \sum_{i=0,\ldots,n} d_i k^i$ which, seen in Horner style:

$$k(k(\ldots k(kd_n + d_{n-1}) + d_{n-2}) \ldots) + d_1) + d_0$$

is a composition of applications $S_{d_0} \circ S_{d_1} \circ \ldots \circ S_{d_n}(0)$ where $S_d : x \mapsto kx + d$.

If a representation uses digits $d \in D$ then it corresponds to the algebra generated by 0 and the S_d's where $d \in D$.

i. The k-adic representation uses digits $1, 2, \ldots, k$ and corresponds to a free algebra built up from one generator and k unary functions.

ii. The usual k-ary representation uses digits $0, 1, \ldots, k-1$ and corresponds to the quotient of a free algebra built up from one generator and k unary functions, namely 0 and the S_d's where $d = 0, 2, \ldots, k-1$, by the relation $S_0(0) = 0$.

iii. Avizienis base k representation uses digits $-k+1, \ldots, -1, 0, 1, \ldots, k-1$, (it is a much redundant representation used to perform additions without carry propagation) and corresponds to the quotient of the free algebra built up from one generator a and $2k-1$ unary functions, namely 0 and the S_d's where $d = -k+1, \ldots, -1, 0, 1, \ldots, k-1$, by the relations $\forall x\ (S_{-k+i} \circ S_{j+1}(x) = S_i \circ S_j(x))$ where $-k < j < k-1$ and $0 < i < k$.

3) $R : \mathbf{N}^4 \to \mathbf{N}$ such that $R(x, y, z, t) = x^2 + y^2 + z^2 + t^2$ is a representation based on Lagrange's four squares theorem

4) $R : Prime^{\leq 7} \to \mathbf{N}$ such that $R(x_1, \ldots, x_i) = x_1 + \ldots + x_i$ (with $i \leq 7$) is a representation based on Schnirelman's theorem (as improved by Ramar, 1995) which insures that every number is the sum of at most 7 prime numbers.

Besides such number theoretic representations of integers, we shall consider classical set theoretic representations involving higher order objects.

Example 2.2.
1) **Church's representation.**
Integers are viewed as function iterators (which are type 2 objects) $f \mapsto f^{(n)}$ where $f^{(0)} = Id$ and $f^{(n+1)} = f^{(n)} \circ f$. Thus, C is the class containing all functional sets $(X \to X)^{X \to X}$ (cf. Notations 1.1) and R is defined on the proper subclass of C constituted of functionals which are finite iterators on some $X \to X$ and $R(F) = n$ if and only if $F(f) = f^{(n)}$ for all $f : X \to X$.

1bis) **Z-Church's representation.**
Negative iterators $f \mapsto f^{(-n)}$ are defined as follows:
i. $Domain(f^{(-n)}) = Range(f^{(n)})$
ii. $f^{(-n)}(x) = y$ if y is the smallest such that $f^{(n)}(y) = x$ (relative to some fixed well-order on X)
The definitions of C and R are analog to that in point 1.

2) **Cardinal representation.**
Integers are viewed as equivalence classes of sets relative to cardinal comparison. Thus, C is the class of all sets and R is defined on the proper subclass

of C constituted of finite sets and $R(X)$ is the cardinal of the set X.

3) **Z-Cardinal representation.**
Relative integers are viewed as differences of natural integers which are themselves viewed via cardinal representation. Thus, C is the class of all pairs of sets and R is defined on the proper subclass of C constituted of finite sets and $R(X, Y)$ is the difference of the cardinals of X and Y.

4) **Ordinal representation.**
Integers are viewed as equivalence classes of totally ordered sets. Thus, C is the class of totally ordered sets and R is defined on the proper subclass of C constituted of finite totally ordered sets and $R(X)$ is the order type of X.

2.2 Formal representations of integers

A formal representation of an integer n is a finite object (in general a word) which describes some characteristic property of n or some abstract object which characterizes n. In fact, each particular representation is really a choice made in order to access special operations or stress special properties of integers.
The computer science (or recursion theoretic) point of view brings an objection to the consideration of abstract sets, functions and functionals as we did in Example 2.2:

- *We cannot apprehend abstract sets, functions and functionals but solely programs to compute them (if they are computable in some sense).*
- *Moreover, programs dealing with sets, functions and functionals have to go through some intensional representation of these objects in order to be able to compute with such objects.*

Thus, to get effectiveness, we turn from set theory to recursion theory and "effectivize" abstract sets:

- sets of integers will be recursively enumerable (r.e.), i.e. domains of partial recursive functions,
- functions on integers will be partial recursive,
- functionals will be partial recursive in the sense of higher type recursion theory (cf. usual textbooks [9] or [8]).

Though abstract representations are quite natural and conceptually simple, their effectivized versions are quite complex: the sets of programs computing objects in their domains involve levels 2 or 3 of the arithmetical hierarchy. In particular, *such representations are not all Turing reducible one to the other.* In the sequel we shall only consider type ≤ 2 representations. In order to get the adequate notion of recursion-theoretic representations of integers, we have to review some higher order recursion concepts.

2.3 Effectivization of sets of type ≤ 2 objects

First, we recall the notion of type 2 recursion that we shall use.

Definition 2.2 (Effective operations).
Let X, Y, Z, T be type 0 spaces (i.e. $\mathbf{N}$, $\mathbf{N}^k$, $\{0,1\}$,...). We denote $PR(X \to Y)$ the set of partial recursive functions $X \to Y$.
An effective operation $F : PR(X \to Y) \to PR(Z \to T)$ or $F : PR(X \to Y) \to Z$ is an operation which can be defined via partial recursive operations on the Gödel numbers of partial recursive functions. In other words, letting $U^{PR(X\to Y)}$ denote a partial recursive enumeration of $PR(X \to Y)$, there exists f such that the following diagram commutes

$$\begin{array}{ccc} PR(X \to Y) & \xrightarrow{F} & PR(Z \to T) \\ {\scriptstyle U^{PR(X\to Y)}}\big\uparrow & & \big\uparrow{\scriptstyle U^{PR(Z\to T)}} \\ \{0,1\}^* & \xrightarrow{f} & \{0,1\}^* \end{array}$$

We denote $Eff((X \to Y) \to (T \to Z))$ (resp. $Eff((X \to Y) \to Z)$) the family of effective operations from $PR(X \to Y)$ into $PR(Z \to T)$ (resp. into Z).

Let's recall the following fact.

Theorem 2.1 (Myhill & Shepherdson, 1955).
Effective operations
$$F : PR(X \mapsto Y) \to Z \quad (\textit{or } F : PR(X \mapsto Y) \to PR(Z \mapsto T))$$
are exactly the restrictions of effectively continuous functionals
$$F : Y^X \to Z \quad (\textit{or } F : Y^X \to T^Z)$$
in the sense of Uspenskii, 1955 (cf. Rogers [9], or Odifreddi [8]).

Definition 2.3 (Effectivization of higher type sets).
1) The effectivization of the type 1 sets Y^X and $X \to Y$ is the subset $PR(X \to Y)$ of partial recursive functions.
2) The effectivization of the type 2 set $(T^Z)^{Y^X}$ (resp. Z^{Y^X}) is the subset of effective operations $PR(X \to Y) \to PR(Z \to T)$ (resp. $PR(X \to Y) \to Z$).

We can now define *strongly universal* partial recursive functions and *strongly universal* effective operations.

Definition 2.4 (Universal enumerations).
Let X, Y, Z, T be type 0 sets and let $\mathcal{E}$ be X or $PR(X \to Y)$ (resp. $Eff((X \to Y) \to (Z \to T))$).

1) A partial recursive function (resp. effective operation) $U : \{0,1\}^* \to \mathcal{E}$ is *universal* if there is a recursive function $\mathtt{comp} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that if we denote $U_{\mathtt{e}}$ the function such that $U_{\mathtt{e}}(\mathtt{p}) = U(\mathtt{comp}(\mathtt{e},\mathtt{p}))$,

then the family $(U_{\mathtt{e}})_{\mathtt{e}\in\{0,1\}^*}$ is an enumeration of the class of partial recursive functions (resp. effective operations) from $\{0,1\}^*$ to $\mathcal{E}$.
Intuition: Words in $\{0,1\}^*$ are seen as programs. A partial recursive function (resp. effective operation) $\{0,1\}^* \to \mathcal{E}$ maps a program to the object it computes (which lies in $\mathcal{E}$). Function $\mathtt{p} \mapsto \mathtt{comp}(\mathtt{e},\mathtt{p})$ is therefore seen as a compiler.
We say that $\mathtt{e}$ is a Gödel number for $F \in \mathcal{E}$ if $F = U_{\mathtt{e}}$.

2) U is *strongly universal* if it is universal and for each index $\mathtt{e}$, there is a constant $c(\mathtt{e})$ such that for all $\mathtt{p} \in \{0,1\}^*$, we have

$$|\mathtt{comp}(\mathtt{e},\mathtt{p})| \leq |\mathtt{p}| + c(\mathtt{e})$$

The following theorem is a classical result of recursion theory which is crucial for the definition of Kolmogorov complexity in §2.5.

Theorem 2.2.
There exists a strongly universal partial recursive function (resp. effective operation).
Moreover, one can suppose that $\mathtt{comp}$ *is the pairing function* $\langle , \rangle$ *defined by*

$$\langle \epsilon, \mathtt{p}\rangle = 1\mathtt{p} \ , \ \ \langle \mathtt{e}_1\mathtt{e}_2 \cdots \mathtt{e}_n, \mathtt{p}\rangle = 0\mathtt{e}_1 0\mathtt{e}_2 \cdots 0\mathtt{e}_n 1\mathtt{p}$$

which satisfies the equality

$$|\langle \mathtt{e}, \mathtt{p}\rangle| = |\mathtt{p}| + 2\ |\mathtt{e}| + 1$$

2.4 Recursion-theoretic representations of integers

Finally, we are in a position to introduce the wanted definition.

Definition 2.5 (Recursion-theoretic representations).
A recursion theoretical representation of $\mathbf{N}$ (resp. $\mathbf{Z}$) is any *surjective* function $\rho : C \to \mathbf{N}$ (resp. $\rho : C \to \mathbf{Z}$) where C is the effectivization $\mathcal{E}$ of some higher type set.

We use Example 2.2 to illustrate the effectivization processes described in §2.3.

Example 2.3.
1) **Effective Church and Z-Church representations.**
Iterators $It_n : PR(\mathbf{N} \to \mathbf{N}) \to PR(\mathbf{N} \to \mathbf{N})$ of partial recursive functions are inductively defined as follows for $n \in \mathbf{N}$:
i. $It_0(f) = f$
ii. $It_{n+1}(f) = It_n(f) \circ f$

Negative iterators $It_{-n} : PR(\mathbf{N} \to \mathbf{N}) \to PR(\mathbf{N} \to \mathbf{N})$ are defined as follows:
iii. $It_{-n}(f)$ has domain $Range(It_n(f)$
iv. $It_{-n}(f)(x) = y$ if $It_n(f)(y) = x$ and $It_n(f)(y)$ is the first halting computation among all computations $It_n(f)(0), It_n(f)(1), \ldots$.

We let $Church^{\mathbf{N}} : Eff((\mathbf{N} \to \mathbf{N}) \to (\mathbf{N} \to \mathbf{N})) \to \mathbf{N}$ be so that
$Church^{\mathbf{N}}(F) = n$ if $F = It_n$ for some $n \in \mathbf{N}$, otherwise undefined

and $Church^{\mathbf{Z}} : Eff((\mathbf{N} \to \mathbf{N}) \to (\mathbf{N} \to \mathbf{N})) \to \mathbf{Z}$ be so that
$Church^{\mathbf{Z}}(F) = n$ if $F = It_n$ for some $n \in \mathbf{Z}$, otherwise undefined

2) **Effective cardinal and Z-cardinal representations.**
We let $card^{\mathbf{N}} : PR(\mathbf{N} \to \mathbf{N}) \to \mathbf{N}$ be so that
i. $card^{\mathbf{N}}(f)$ is defined if and only if $domain(f)$ is finite
ii. $card^{\mathbf{N}}(f) = card(domain(f))$

We let $card^{\mathbf{Z}} : PR(\mathbf{N} \to \mathbf{N}) \times PR(\mathbf{N} \to \mathbf{N}) \to \mathbf{Z}$ be so that
iii. $card^{\mathbf{Z}}(f, g)$ is defined if and only if f, g have finite domains
iv. $card^{\mathbf{Z}}(f, g) = card(domain(f)) - card(domain(g))$

3) **Effective ordinal representation.**
We let $ord^{\mathbf{N}} : PR(\mathbf{N}^2 \to \mathbf{N}) \to \mathbf{N}$ be so that
i. $ord^{\mathbf{N}}(f)$ is defined if and only if the quotient order associated to the transitive closure of $domain(f)$ is finite
ii. $ord^{\mathbf{N}}(f)$ is the order type of this quotient order.

The following result measures the syntactical complexity of the domain and the graph of the functionals $Church^{\mathbf{N}}, Church^{\mathbf{Z}}, card^{\mathbf{N}}, card^{\mathbf{Z}}, ord$ in terms of the associated index sets.

Proposition 2.1 (Syntactical complexity of representations).
1) **Church representations.**
The set of pairs (e, n) *such that* $n \in \mathbf{N}$ *(resp.* $n \in \mathbf{Z}$*) and* e *is a Gödel number for the iteration functional* It_n *is* Π^0_2*-complete.*
The set of Gödel numbers of effective functionals in the domain of $Church^{\mathbf{N}}$ *(resp.* $Church^{\mathbf{Z}}$*) is* Σ^0_3*-complete.*

2) **Cardinal representations.**
The set of pairs (e, n) *such that* $n \in \mathbf{N}$ *(resp.* $n \in \mathbf{Z}$*) and* e *is the Gödel number of a partial recursive function the domain of which is finite with* n *elements is* Σ^0_2*-complete.*
The set of Gödel numbers of partial recursive functions with finite domains is Σ^0_2*-complete.*

3) **Ordinal representation.**
The set of pairs (e, n) *such that* $n \in \mathbf{N}$ *and* e *is the Gödel number of a partial recursive function* $\mathbf{N}^2 \to \mathbf{N}$ *such that the quotient order associated to the transitive closure of* $domain(f)$ *is finite with* n *elements is* Σ^0_3*-complete.*
The set of Gödel numbers of partial recursive functions such that the quotient order associated to the transitive closure of $domain(f)$ *is finite is* Σ^0_3*-complete.*

2.5 Kolmogorov complexity and representations of integers

The usual definition of Kolmogorov complexity can be extended to any recursion-theoretic representation of integers.

Definition 2.6 (Kolmogorov complexity relative to a representation). Let $\mathcal{E}$ be the effectivization of some higher type set and let $\rho : \mathcal{E} \to \mathbf{N}$ (resp. $\rho : \mathcal{E} \to \mathbf{Z}$) be a recursion-theoretic representation of integers. Considering the diagram

$$\{0,1\}^* \xrightarrow{U^{\mathcal{E}}} \mathcal{E} \xrightarrow{\rho} \mathbf{N}$$

where $U^{\mathcal{E}}$ is some strongly universal enumeration of $\mathcal{E}$, Kolmogorov complexity $K_\rho : \mathbf{N} \to \mathbf{N}$ is defined as

$$K^{\mathbf{N}}_{\rho}(n) = \min\{|p| \ : \rho(U^{\mathcal{E}}(p)) = n\}$$

The definition of $K_\rho : \mathbf{Z} \to \mathbf{N}$ is analog.

Theorem 2.2 implies an invariance theorem for strongly universal enumerations. Which insures that the above definition does not depend (up to a constant) of the particular choice of the strongly universal enumeration $U^{\mathcal{E}}$ of $\mathcal{E}$.

Remark 2.2. The domain of $\rho \circ U^{\mathcal{E}}$ is not recursively enumerable in general (cf. Prop. 2.1).

3 Main Theorem

3.1 Main theorem

Reconsidering the answer to ($Q2$) given after Def. 1.1, the main theorem of this paper insures the following.

- Kolmogorov complexity is much dependent on the chosen higher order effective representations of integers,
- The Kolmogorov complexities associated to representations of Example 2.3 constitute a hierarchy which coincides with that obtained with infinite computations and relativization to the jumps (cf. Prop. 1.1).

Thus, Kolmogorov complexity measures the complexity of higher order effective representations and allows to classify them.

Theorem 3.1 (Main Theorem).
Let $K^{\mathbf{N}}_{Church}, K^{\mathbf{Z}}_{Church}, K^{\mathbf{N}}_{card}, K^{Z\mathbf{N}}_{card}, K_{ord}$ be the Kolmogorov complexities associated to the effective versions of the higher order representations described in Example 2.3. Then

$$K^{\mathbf{N}}_{Church} =_{\mathrm{ct}} K^{\mathbf{Z}}_{Church} \restriction \mathbf{N} =_{\mathrm{ct}} K$$

$$K^{\mathbf{N}}_{card} =_{\rm ct} K^{\infty}$$
$$K^{\mathbf{Z}}_{card} \upharpoonright \mathbf{N} =_{\rm ct} K'$$
$$K^{\mathbf{N}}_{ord} =_{\rm ct} K'^{\infty}$$

So that we have

$$K^{\mathbf{N}}_{Church} =_{\rm ct} K^{\mathbf{Z}}_{Church} \upharpoonright \mathbf{N} >_{\rm ct} K^{\mathbf{N}}_{card} >_{\rm ct} K^{\mathbf{Z}}_{card} \upharpoonright \mathbf{N} >_{\rm ct} K_{ord}$$

3.2 Representations ρ such that $K_\rho = K$

The following theorem gives simple sufficient conditions on representations ρ so that the associated Kolmogorov complexity K_ρ be equal, up to a constant, to usual Kolmogorov complexity K.
In particular, these conditions will apply to Church's representations.

Theorem 3.2.
Let $\mathcal{E}$ be $PR(X \to Y)$ or $Eff((X \to Y) \to (Z \to T)$ and $U^{\mathcal{E}} : \{0,1\}^* \to \mathcal{E}$ be strongly universal. Let $\rho : \mathcal{E} \to \mathbf{N}$ (resp. $\rho : \mathcal{E} \to \mathbf{Z}$) be a representation of integers.
Consider the following conditions:

(*) $\rho \circ U^{\mathcal{E}}$ is the restriction to its domain of some partial recursive function $f : \{0,1\}^* \to \mathbf{N}$ ($f : \{0,1\}^* \to \mathbf{Z}$).

(**) ρ is effectively surjective*: there exists a* total recursive *function $g : \mathbf{N} \to \{0,1\}^*$ (resp. $g : \mathbf{Z} \to \{0,1\}^*$) such that $\rho(U^{\mathcal{E}}(g(n))) = n$ for all $n \in \mathbf{N}$ (resp. $n \in \mathbf{N}$)*

1) Condition (*) implies $K \leq_{\rm ct} K_\rho$.

2) Condition (**) implies $K_\rho \leq_{\rm ct} K$.

Proof.
1) Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program such that $\rho(U^{\mathcal{E}}(p)) = n$. Then $K_\rho(n) = |p|$. Condition (*) implies $f(p) = n$. Therefore, viewing f as a programming language and using Kolmogorov's invariance theorem there is a constant c independent of n such that
$$K(n) \leq K_f(n) + c \leq |p| + c = K_\rho(n) + c$$
Thus, $K \leq_{\rm ct} K_\rho$.

2) Let $U^{\mathbf{N}} : \{0,1\}^* \to \mathbf{N}$ be a strongly universal enumeration of $\mathbf{N}$. The strong universality of $U^{\mathcal{E}}$ insures that there exists e such that

$$\forall p \ \ U^{\mathcal{E}}(g(U^{\mathbf{N}}(p))) = U^{\mathcal{E}}(\langle \mathsf{e}, p \rangle)$$

Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program such that $U^{\mathbf{N}}(p) = n$. Then $K(n) = |p|$.

Condition (**) implies $\rho(U^{\mathcal{E}}(g(U^{\mathbf{N}}(p)))) = n$. Thus, $\rho(U^{\mathcal{E}}(\langle \mathsf{e}, p\rangle)) = n$, whence (using Thm 2.2)

$$K_\rho(n) \leq |\langle \mathsf{e}, p\rangle| = |p| + 2|\mathsf{e}| + 1 = K(n) + 2|\mathsf{e}| + 1$$

and therefore $K_\rho \leq_{\mathrm{ct}} K$. □

Corollary 3.1.

$$K^{\mathbf{N}}_{Church} =_{\mathrm{ct}} K^{\mathbf{Z}}_{Church} \restriction \mathbf{N} =_{\mathrm{ct}} K$$

Proof.
We show that conditions (*) and (**) of Theorem 3.2 are satisfied for the ρ associated to $K^{\mathbf{N}}_{Church}$. The argument also applies with straightforward modifications to $K^{\mathbf{Z}}_{Church} \restriction \mathbf{N}$.
To get condition (**), just design a program for functional It_n.
As for condition (*), let $\mathcal{E} = Eff((\mathbf{N} \to \mathbf{N}) \to (\mathbf{N} \to \mathbf{N}))$ and $Succ : \mathbf{N} \to \mathbf{N}$ be the successor function and define $f\{0,1\}^* \to \mathbf{N}$ as follows:

$$f(p) = U^{\mathcal{E}}(p)(Succ)(0)$$

If $\rho(U^{\mathcal{E}}(p)) = n$ then $U^{\mathcal{E}}(p) = It_n$ so that $U^{\mathcal{E}}(p)(Succ)$ is the function $x \mapsto x + n$ and $f(p) = U^{\mathcal{E}}(p)(Succ)(0) = n$. □

Remark 3.1.
1) Conditions (*) and (**) both hold trivially in case $\rho \circ U^{\mathcal{E}}$ is a partial recursive function or an effective functional.

2) Theorem 3.2 relativizes to computability with an oracle.

3.3 $K^{\mathbf{N}}_{card}$ and K^∞

Theorem 3.3.

$$K^{\mathbf{N}}_{card} =_{\mathrm{ct}} K^\infty$$

Proof.
1) $\mathcal{E} = PR(\mathbf{N} \to \mathbf{N})$. Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program which outputs n in unary through a possibly infnite computation. Then $K^\infty(n) = |p|$. Define $h : \{0,1\}^* \to \{0,1\}^*$ such that $h(p)$ is the following program for a partial recursive function $\varphi_{h(p)} : \mathbf{N} \to \mathbf{N}$:

$$\varphi_{h(p)}(t) = \texttt{IF} \text{ at step } t \text{ program } p \text{ outputs } 1 \;\texttt{THEN}\; 1 \;\texttt{ELSE}\; \text{undefined}$$

Clearly, $card^{\mathbf{N}}(U^{\mathcal{E}}(h(p)) = n$. The strong universality of $U^{\mathcal{E}}$ insures that there exists e such that

$$\forall p \;\; U^{\mathcal{E}}(h(p)) = U^{\mathcal{E}}(\langle \mathsf{e}, p\rangle)$$

This leads to $K^{\mathbf{N}}_{card}(n) \leq K^\infty(n) + 2|\mathsf{e}| + 1$, whence $K^{\mathbf{N}}_{card} \leq_{\mathrm{ct}} K^\infty$.

2) Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program such that $card^{\mathbf{N}}(U^{\mathcal{E}}(p)) = n$. Define $h : \{0,1\}^* \to \{0,1\}^*$ such that $h(p)$ behaves as follows:

- $h(p)$ emulates some dovetailing of all computations $U^{\mathcal{E}}(p)(i)$ for $i = 0, 1, 2, \ldots$,
- each time some computation $U^{\mathcal{E}}(p)(i)$ halts (i.e. a new point i is proved to be in the domain of $U^{\mathcal{E}}(p)$) then $h(p)$ output 1.

It is clear that $h(p)$ outputs n in unary. Thus, $K^{\infty}(n) \leq |h(p)|$. But there is some constant c such that $|h(p)| = |p| + c$, whence $K^{\infty}(n) \leq |p| + c$ and therefore $K^{\infty} \leq_{\mathrm{ct}} K^{\mathbf{N}}_{card}$. □

3.4 $K^{\mathbf{Z}}_{card} \restriction \mathbf{N}$ and K'

Theorem 3.4.

$$K^{\mathbf{Z}}_{card} \restriction \mathbf{N} =_{\mathrm{ct}} K'$$

Proof.
Now $\mathcal{E} = (PR(\mathbf{N} \to \mathbf{N}))^2$ and $U^{\mathcal{E}}$ is a pair of functions $(U^{\mathcal{E}}_1, U^{\mathcal{E}}_2)$.
1) Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program such that $card^{\mathbf{Z}}(U^{\mathcal{E}}(p)) = card^{\mathbf{N}}(U^{\mathcal{E}}_1(p)) - card^{\mathbf{N}}(U^{\mathcal{E}}_2(p)) = n$. Define $h : \{0,1\}^* \to \{0,1\}^*$ such that $h(p)$ is a program which uses oracle $\emptyset'$ and behaves as follows:

- $h(p)$ emulates some dovetailing of all computations $U^{\mathcal{E}}_1(p)(i), U^{\mathcal{E}}_2(p)$ for $i = 0, 1, 2, \ldots$,
- At each computation step, $h(p)$ asks the oracle whether there is still some computation that will halt. If the answer is "NO" then $h(p)$ halts and outputs the difference of the number of points i's on which $U^{\mathcal{E}}_1(p)$ has been checked to converge and that for $U^{\mathcal{E}}_2(p)$

It is clear that $h(p)$ outputs n. Thus, $K'(n) \leq |h(p)|$. But there is some constant c such that $|h(p)| = |p| + c$, whence $K'(n) \leq |p| + c$ and therefore $K' \leq_{\mathrm{ct}} K^{\mathbf{Z}}_{card} \restriction \mathbf{N}$.

2) Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program using oracle $\emptyset'$ which outputs n in unary. Then $K'(n) = |p|$. To emulate computations using oracle $\emptyset'$, we shall use Chaitin's harmless overshoot technique [3]:

- If an $\exists x \ldots$ assertion is true then a computation loop will check that it is true.
- However, if it is false, there is no way to check it in finite time.
- Whence the strategy to systematically answer "NO" to each $\exists x \ldots$ question and then check this answer via a computation loop.

- Every false answer "NO" will be proved false at some time during this loop, giving a possibility to correct it.

- Every computation using oracle $\emptyset'$ which halts uses finitely many questions to the oracle. The above strategy will therefore be corrected only finitely many times so that it eventually leads for a fair emulation.

Define $h : \{0,1\}^* \to \{0,1\}^*$ such that $U_1^{\mathcal{E}}(h(p))$ and $U_2^{\mathcal{E}}(h(p))$ behave as follows:

1. $U_1^{\mathcal{E}}(h(p))$ emulates p and whenever p outputs 1 then $U_1^{\mathcal{E}}(h(p))$ will converge on t where t is the current computation step.

2. Each time p asks a question to oracle $\emptyset'$ of the form $\exists x \ldots$? then $U_1^{\mathcal{E}}(h(p))$ answers "NO".

3. Cautiously, $U_1^{\mathcal{E}}(h(p))$ checks each of its oracle answers starting a computation loop which will halt only if the right answer was "YES" (instead of "NO").

4. In case some answer was false, then $U_1^{\mathcal{E}}(h(p))$ restarts the whole emulation of p (correcting its past answers) and $U_2^{\mathcal{E}}(h(p))$ will converge on a set of points in bijection with that on which $U_1^{\mathcal{E}}(h(p))$ was made converging before the answer was recognized to be false.

Corrections brought in point 4 make the final difference

$$card^{\mathbf{N}}(U_1^{\mathcal{E}}(h(p))) - card^{\mathbf{N}}(U_2^{\mathcal{E}}(h(p)))$$

equal to n.
The strong universality of $U^{\mathcal{E}}$ insures that there exists $\mathtt{e}$ such that

$$\forall p \ (U_1^{\mathcal{E}}(h(p)) = U^{\mathcal{E}}(\langle \mathtt{e}_1, p\rangle) \ \wedge \ U_2^{\mathcal{E}}(h(p)) = U^{\mathcal{E}}(\langle \mathtt{e}_2, p\rangle))$$

This leads to $K^{\mathbf{Z}}_{card}(n) \leq K'(n) + 2|\mathtt{e}| + 1$, whence $K^{\mathbf{Z}}_{card} \restriction \mathbf{N} \leq_{\mathrm{ct}} K'$. □

3.5 K_{ord} and K'^{∞}

Theorem 3.5.

$$K_{ord} =_{\mathrm{ct}} K'^{\infty}$$

Proof.
Now $\mathcal{E} = PR(\mathbf{N}^2 \to \mathbf{N})$.
1) Let $n \in \mathbf{N}$ and let $p \in \{0,1\}^*$ be a minimal length program such that $ord(U^{\mathcal{E}}(p)) = n$. Define $h : \{0,1\}^* \to \{0,1\}^*$ such that $h(p)$ uses oracle $\emptyset'$ and behaves as follows:

- $h(p)$ initializes a set X of integers to $\emptyset$.

- **Step 0.** $h(p)$ checks if there is some pair with 0 as a component which is in the domain of $U^{\mathcal{E}}(p)$. If so it outputs a 1 and puts 0 in the set X.

- **Step $t > 0$.** $h(p)$ checks if there is some chain (constituted of pairs in the domain of $U^{\mathcal{E}}(p)$) containing t and all elements of X. If so it outputs a 1 and puts t in the set X.

It is clear that, through an infinite computation, $h(p)$ outputs the unary representation of the order type of the transitive closure of the domain of $U^{\mathcal{E}}(p)$ in case it is a finite ordered type. Thus, $K'^{\infty}(n) \leq |h(p)|$. But there is some constant c such that $|h(p)| = |p| + c$, whence $K'^{\infty}(n) \leq |p| + c$ and therefore $K'^{\infty} \leq_{\text{ct}} K_{ord}$.

2) Let $n \in \mathbf{N}$, $n > 0$ and let $p \in \{0,1\}^*$ be a minimal length program using oracle $\emptyset'$ which outputs n in unary through an infinite computation. Then $K'^{\infty}(n) = |p|$. We shall again use Chaitin's harmless overshoot technique. Define $h : \{0,1\}^* \to \{0,1\}^*$ such that $U^{\mathcal{E}}(h(p))$ behaves as follows:

1. $U^{\mathcal{E}}(h(p))$ initializes a set X of integers to $\{0\}$. This set X will always be finite.

2. $U^{\mathcal{E}}(h(p))$ emulates p and whenever p outputs 1 then a new point k is added to X and $U^{\mathcal{E}}(h(p))$ will converge on every pair (x, k) where $x \in X$.

3. Each time p asks a question to oracle $\emptyset'$ of the form $\exists x \ldots$? then $U^{\mathcal{E}}(h(p))$ answers "NO".

4. Cautiously, $U^{\mathcal{E}}(h(p))$ checks each one of its oracle answers starting computation loops which will halt only if some right answer was "YES" (instead of "NO").

5. In case some answer was false, then $U^{\mathcal{E}}(h(p))$ restarts the whole emulation of p (correcting its past answers) and $U^{\mathcal{E}}(h(p))$ will converge on all pairs $(x, y) \in X^2$ and the set X is reinitialized to $\{0\}$.

Corrections brought in point 5 make the final transitive closure of the domain of $U^{\mathcal{E}}(h(p))$ a preordered set the quotient of which has exactly n points. The strong universality of $U^{\mathcal{E}}$ insures that there exists e such that

$$\forall p \ (U^{\mathcal{E}}(h(p)) = U^{\mathcal{E}}(\langle \mathsf{e}, p \rangle))$$

This leads to $K_{ord}(n) \leq K'^{\infty}(n) + 2|\mathsf{e}| + 1$, whence $K_{ord} \leq_{\text{ct}} K'^{\infty}$. $\square$

References

[1] V. Becher, S. Daicz, and G. Chaitin. A highly random number. In *Combinatorics, Computability and Logic: Proceedings of the Third Discrete Mathematics and Theoretical Computer Science Conference (DMTCS'01)*, pages 55–68. Springer-Verlag, 2001.

[2] G. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM*, 22:329–340, 1975. Available on Chaitin's home page.

[3] G. Chaitin. Information theoretic characterizations of infinite strings. *Theoret. Comput. Sci.*, 2:45–48, 1976. Available on Chaitin's home page.

[4] M. Ferbus-Zanda and S. Grigorieff. Is randomnes native to computer science? *Bull. EATCS*, 74:78–118, 2001. Available on the author's home page.

[5] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

[6] L. Levin. On the notion of random sequence. *Soviet Math. Dokl.*, 14(5):1413–1416, 1973.

[7] M. Li and P. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer, 1997 (2d edition).

[8] P. Odifreddi. *Classical Recursion Theory*, volume 125. North-Holland, 1989.

[9] H. Rogers. *Theory of recursive functions and effective computability.* McGraw-Hill, 1967.

[10] A. Shen. Kolmogorov complexity and its applications. *Lecture Notes, Uppsala University, Sweden*, pages 1–23, 2000. http://www.csd.uu.se/˜vorobyov/ Courses/KC/2000/all.ps.

[11] R.M. Solovay. On random R.E. sets. In A.I. Arruda and al., editors, *Non-classical Logics, Model theory and Computability*, pages 283–307. North-Holland, 1977.